Exact Reductions and Lower Bounds to Reduction Efficiency in Cryptology

Margus Niitsoo

(joint work with Ahto Buldas and Aivo Jürgenson)

February 1, 2009

Exact Reductions and Upper Bounds to Reduction Efficiency in Cryptology

Margus Niitsoo

(joint work with Ahto Buldas and Aivo Jürgenson)

February 1, 2009

Outline

1. What are cryptographic reductions
2. What has been studied
3. What we are doing

Cryptographic practice

- We (in general) do not know how to do cryptology!

Cryptographic practice

- We (in general) do not know how to do cryptology!
- That does not stop us!

Cryptographic practice

- We (in general) do not know how to do cryptology!
- That does not stop us!
- We just assume we have gotten some things right
  - And then show how to do the other things based on them.

Collision resistant function

- Assume we have an h for which it is hard to find two inputs $x \neq x'$ such that $h(x) = h(x')$.
- Such a pair is called a collision.
- Such functions are assumed to exist

Example: Merkle-Damgård construction

How to construct a collision-resistant

$$h' : \{0,1\}^{4n} \rightarrow \{0,1\}^n$$

from a collision-resistant

$$h : \{0,1\}^{2n} \rightarrow \{0,1\}^n$$

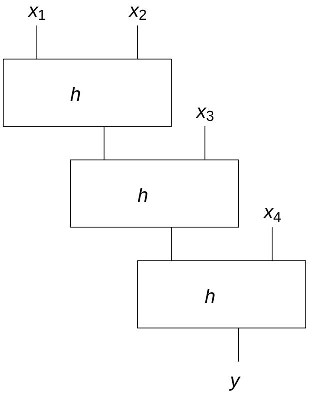
Example: Merkle-Damgård construction

How to construct a collision-resistant

$$h' : \{0,1\}^{4n} \to \{0,1\}^n$$

from a collision-resistant

$$h : \{0,1\}^{2n} \to \{0,1\}^n$$



Reductions in cryptology

To show that one primitive can be constructed from another

- You need to show a construction that builds one from another
- You also need to prove that the new construction is secure
  - To do that, one usually shows the contrapositive, that is, if it is not secure, then the original primitive used also is not.
  - That is done by constructing an explicit adversary

So a reduction in cryptology

- Has a construction that realizes it
- Has a security construction
  - That takes an adversary for the new construction
  - And uses it to break the old primitive

Interesting questions

Given two primitives X and Y,

- Does there exist a reduction from one to the other?
- If so, how efficient is it?

Reductions we know exist

- Given a one-way function, we can do all of secret-key cryptology
- Given a trapdoor one-way function, we can do all of public-key cryptology
- More specific reductions are known but are too numerous to mention

Reductions we know **not to exist**

- We know that we cannot create public-key primitives from secret-key primitives in a purely black-box way (Rudich-Impagliazzo 89)
- Many others are known
  - Collision-resistant functions cannot be constructed from a time-stamping scheme (Buldas, Jürgenson 07)

Reductions we know not to be efficient

- When is a reduction efficient?

Reductions we know not to be efficient

- When is a reduction inefficient?

Reductions we know not to be efficient

- When is a reduction inefficient?
- Usual answer: If it uses the original primitive prohibitively often

Reductions we know not to be efficient

- When is a reduction inefficient?
- Usual answer: If it uses the original primitive prohibitively often
  - In this context, inefficient means a linear number of calls to the original primitive

Reductions we know not to be efficient

- When is a reduction inefficient?
- Usual answer: If it uses the original primitive prohibitively often
  - In this context, inefficient means a linear number of calls to the original primitive
- Pseudo-random generators and SKE cannot be efficiently implemented using just one-way permutations in a black-box way
- PKE and Signature schemes cannot be efficienttly constructed from trapdoor one-way permutations.

A reduction we know rather well

- Most of the research of our subgorup has been on time-stamping
- We have a construction of tree-based time-stamping using collision-resistant functions that is used in practice
- It has been proven secure

A reduction we know rather well

- Most of the research of our subgorup has been on time-stamping
- We have a construction of tree-based time-stamping using collision-resistant functions that is used in practice
- It has been proven secure
- However - we want to know if we can prove it even more secure.

Adversary efficiency

- We model the efficency of the adversary with its time/success ratio
  - The smaller the ratio, the better the adversary
  - Conversely, the larger the ratio, the more secure the primitive (assuming we have the best possible adversary)

Reduction efficiency

- We say that we have a *power c fully black-box reduction* between two primitives if the adversary construction S guarantees

$$\frac{\textsc{Time}_k(\mathsf{S}^{\mathsf{A},f},f)}{\textsc{Adv}_k(\mathsf{S}^{\mathsf{A},f},f)} \le k^{O(1)} \cdot \left[\frac{\textsc{Time}_k(\mathsf{A},\mathsf{P}^f)}{\textsc{Adv}_k(\mathsf{A},\mathsf{P}^f)}\right]^c \quad .$$

- Essentially, we have a power c reduction if the time/success ratio of the constructed adversary is less than the old time/success ratio raised to power c.
- So the adversary to the original primitive constructed from an adversary to the constructed primitive has to be at least as good as...
  - Smaller is better

Reduction efficiency in practice

- Suppose we have a power-c reduction that gives a construction for secure time-stamping from collision-resistant functions.

Reduction efficiency in practice

- Suppose we have a power-c reduction that gives a construction for secure time-stamping from collision-resistant functions.
- We use that construction on an actual collision-resistant function to get a secure time-stamping function.

Reduction efficiency in practice

- Suppose we have a power-c reduction that gives a construction for secure time-stamping from collision-resistant functions.
- We use that construction on an actual collision-resistant function to get a secure time-stamping function.
- Suppose that the best possible adversary for that collision-resistant function has time/success ratio of r.
- Then the reduction being power-c gives us a guarantee that the best possible adversary against our construction can have a time-success ratio of at most $r^{1/c}$.
  - As otherwise we could use it to construct a better adversary to collision-resistant function than we believe possible.

Reduction efficiency in practice

- Suppose we have a power-c reduction that gives a construction for secure time-stamping from collision-resistant functions.
- We use that construction on an actual collision-resistant function to get a secure time-stamping function.
- Suppose that the best possible adversary for that collision-resistant function has time/success ratio of r.
- Then the reduction being power-c gives us a guarantee that the best possible adversary against our construction can have a time-success ratio of at most $r^{1/c}$.
    - As otherwise we could use it to construct a better adversary to collision-resistant function than we believe possible.
- We are neglecting some constants here so this analysis is approximate.

What we have done

- We prove separation theorems for power-c reductions
  - "No power-c black-box reduction can exist for a given c"
  - Allows to give a lower bound on c for a given reduction
  - Is a lower bound on efficiency, as smaller c means a better reduction.
- As an example of use, we prove a lower bound of 1.5 for constucting tree-based time-stamping from collision-resistant hash functions.

What does it mean?

- Remmember we are interested in how good security guarantees can theoretically be achieved with a black-box reduction in our time-stamping context
- The 'Example' shows that the best reduction can only give us a $c = 1.5$.

What does it mean?

- Remmember we are interested in how good security guarantees can theoretically be achieved with a black-box reduction in our time-stamping context
- The 'Example' shows that the best reduction can only give us a $c = 1.5$.
- What I haven't told you yet is that we actually **Have the reduction with** $c = 1.5$.
- Our article proves it is optimal!
- This means that we can stop looking for better reductions and that the problem is essentially solved in the best possible way.

What does it mean?

- Remmember we are interested in how good security guarantees can theoretically be achieved with a black-box reduction in our time-stamping context
- The 'Example' shows that the best reduction can only give us a $c = 1.5$.
- What I haven't told you yet is that we actually **Have the reduction with** $c = 1.5$.
- Our article proves it is optimal!
- This means that we can stop looking for better reductions and that the problem is essentially solved in the best possible way.
- However, we plan to publish that article a bit later so please, don't tell anybody about this last slide;)

Thank You!

Thank you for attention! Any questions are welcome!